VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

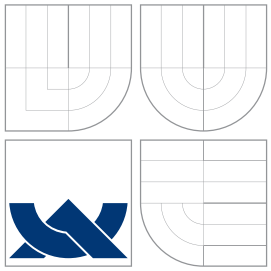
ROZŠÍŘENÁ PRÁVA UŽIVATELŮ SYSTÉMU DOKUWIKI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

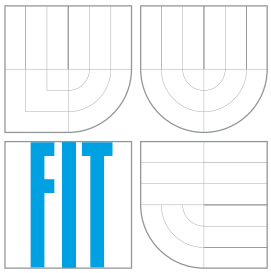
AUTOR PRÁCE
AUTHOR

ONDŘEJ MACHÁČ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ROZŠÍŘENÁ PRÁVA UŽIVATELŮ SYSTÉMU DOKUWIKI

EXTENDING ACCESS CONTROL LIST IN DOKUWIKI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — ONDŘEJ MACHÁČ
AUTHOR

VEDOUCÍ PRÁCE — Ing. ALEŠ SMRČKA, Ph.D.
SUPERVISOR

BRNO 2011

Abstrakt

Tato bakalářská práce má za úkol seznámit nás s wiki systémy, především pak DokuWiki. Zabývá se historií,právními aspekty a možností tvorby zásuvných modulů. Ty jsou také hlavní částí. V prvním případě je navrhnut jednoduchý zásuvný modul pro přidělování práv. Ve druhém případě je rozšířeno zadávání práv a je vytvořena možnost automatického přihlašování uživatelů na základě různých vstupních informací.

Abstract

The goal of this bachelor thesis is to introduce the Wiki systems, especially DokuWiki. It deals with history, legal aspects and possibility to create pluggable modules (alias plugins). Plugins are also main theme of this work. At first, a simple plugin for user access rights is designed. Afterwards is extended plugin's functionality with advanced settings of permissions and is also created the ability of automatic user login based on specific entry informations.

Klíčová slova

ACL, administrátor, bezpečnostní model, dokuwiki, host, kritérium, moderátor, návrh, wiki, zásuvný modul

Keywords

ACL, administrator, security model, dokuwiki, guest, criterion, moderator, draft, wiki, plugin

Citace

Ondřej Macháč: Rozšířená práva uživatelů systému
Dokuwiki, bakalářská práce, Brno, FIT VUT v Brně, 2011

Rozšířená práva uživatelů systému Dokuwiki

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Aleše Smrčky, Ph.D.

.....
Ondřej Macháč
15. května 2011

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce za pomoc, ochotu a čas, který mi věnoval.

Obsah

Kapitola 1

Úvod

V dnešní době je velmi důležité chránit data, která sdílíme na webu před možnými útočníky, kteří by mohli naše data poškodit. Proto je velmi důležité zamezit v přístupu takovýmto lidem a povolit přístup na naše stránky pouze těm, kteří na to mají povolení. Tato bakalářská práce se zabývá tímto zabezpečením.

Na úvod si řekneme něco o bezpečnostních modelech a vysvětlíme si obecně úlohu bezpečnostních modelů, zvláště pak modely založené na rolích. Dále si řekneme něco o wiki systémech, zvláště pak jejich historii, začátky a několik statistických údajů. Další část se zabývá již konkrétním wiki systémem a to je DokuWiki. Tato wiki je určena zvlášť pro online vedení dokumentace v týmech, které spolu spolpracují po celém světě. Navíc má tu výhodu, že je jakkoli rozšiřitelná pomocí zásuvných modulů. Možnost tvorby těchto zásuvných modulů si taktéž osvětlíme.

Jako další bude navrhnut zásuvný modul pro jednuduché přidělování práv. Hlavní a podstatná část práce je věnována návrhu a realizaci zásuvného modulu pro systém DokuWiki, který rozšiřuje jednoduché přidělování práv, zavádí moderátorská práva a automaticky přihlašuje uživatele na základě zadaných vstupních údajů (kritérií).

Kapitola 2

Využité technologie

2.1 Bezpečnostní model založený na rolích (RBAC)

2.1.1 Bezpečnostní modely

Při formulování bezpečnostní politiky musíme popsat entity touto politikou řízené a stanovit pravidla, která tuto politiku tvoří. Je nutné zodpovědět např. tyto otázky:

- jaká je bezpečnostní politika?
- jaká dávat práva, komu, kam?

Toto dělají bezpečnostní modely. Některé bezpečnostní modely zdůrazňují utajení dat, jiné integritu dat. Některé modely jsou statické, jiné dovolují dynamické změny v přístupových právech. Pravidla pro definování politiky určují bezpečnostní modely. Dá se říci, že bezpečnostní modely mohou pomoci při vybudování určité bezpečnostní politiky, jsou jejím konkrétním vyjádřením. Úkolem bezpečnostního modelu je jasně a velmi přesně formulovat bezpečnostní politiku. Dále nutno podotknout, že hodnocení bezpečnosti vyžaduje u některých vyšších stupňů bezpečnosti, např. dle Orange Book i dalších kritérií neformální a posléze i formální model. Modelů byla vyvinuta již celá řada. Některé modely byly v praxi implementovány třeba v poněkud pozměněné podobě, jiné jsou pouze teoretické.

Modely mohou být užitečné např. v těchto příkladech:

- testování částečné bezpečnostní politiky
- jako pomoc pro implementaci bezpečnosti
- rozhodnutí, zda implementace splňuje původní požadavky
- studium těchto modelů vede k lepšímu pochopení, jak a proč chránit systém

Subjekt je aktivní prvek modelu; u některých modelů je to člověk, jinde proces. Aktivní subjekt se snaží o nějakou operaci přístupu k pasivnímu objektu, je zde bezpečnostní monitor, který mu buď přístup povolí nebo zakáže.

Objekt je pasivní prvek např. soubor, adresář, tiskárna, paměť atd. Např. proces může být z tohoto hlediska jednou subjekt a jednou objekt. To samé může být jak objekt tak subjekt, záleží na tom, zda je v aktivní či pasivní roli.

Jedním z dnes často užívaných modelů je Role Based Access Control Model - RBAC. V tomto modelu jsou uživatelé rozlišováni ne podle osobní identity, ale podle přidělené role. Např. v univerzitním prostředí se koncepce modelu RBAC jeví jako vhodná. Role může být např. student oboru Informatika 4. ročník, vyučující předmětu Databáze 1. Těchto vyučujících může být více, každý má stejná práva, jsou zastupitelní, při jakékoliv situaci (nemoc, odchod z pracoviště atd.) se pouze dosadí do role jiný konkrétní člověk. Určitá práva se mohou dědit. Např. vedoucí studijního oddělení má práva všech referentek, které jednotlivě mají práva pouze pro práci s daty studentů svého ročníku, oboru atd., za který odpovídají. Ale navíc má vedoucí nějaká další práva, která jednotlivé referentky nemají. Tento koncept je také velmi vhodný z hlediska ochrany soukromí. Pokud někdo žádá o určitý materiál, není nutné, aby se např. internetem přenášela informace, že uživatel Macháč žádal o to a to. Zcela postačí, pokud žádost zní, že vyučující předmětu Programování (tato informace je ověřena) žádá o přístup k materiálům a to pro akci zápis. To je v souladu s direktivou EU o ochraně soukromí a s celkovými trendy posledních let, kdy se o ochranu soukromí na Internetu začíná více dbát.

2.1.2 Role Based Access Control Model

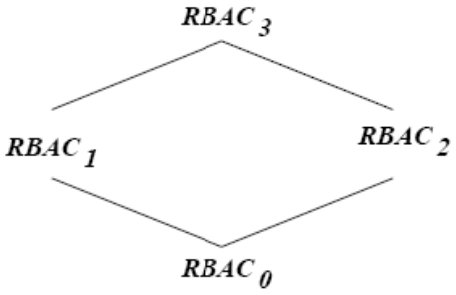
Mezi hybridní politiky (a tedy modely) patří i RBAC politika (a model). Jedná se o model který vyvinul Sandhu a spol. [9], ve vývoji dále pokračovali David Ferraiolo a Richard Kuhn [7]. Tento model uživatele rozděluje podle jejich rolí (lékař, zdravotní sestra, ošetřující lékař, vyučující předmětu, atd.). Jeden uživatel může mít různé role (vyučující předmětů Programovaní v jazyce C, Počítačové sítě, Databáze 1). Každá role je spojena s množinou povolených aktivit, které může uživatel v dané roli konat na zdroji, který je chráněn pomocí nadefinované bezpečnostní politiky. Množina povolených přístupů může být sdílena více rolemi.

Model se velmi hodí pro organizace, kde se mohou jednotliví lidé střídat v rolích (lékař, vyučující), ale kde role jsou poměrně trvalé. Nezáleží na konkrétní osobě, ale na roli, kterou osoba koná a pokud se třeba během semestru změní vyučující, tak pokračuje ve stejné roli. Zejména je tento systém velmi vhodný pro zdravotnictví, vzhledem k pevné hierarchii (sestra, lékař, primář, vrchní sestra atd.), ale i vysokoškolské prostředí má poměrně jasně definovanou hierarchii danou vyučujícími, cvičícími atd. Navíc podobný model je velmi flexibilní a jde upravit podle potřeb dané organizace.

Sandhu a spol.[9] popsali celkem 4 stupně RBAC modelů. RBACo, RBAC1, RBAC2, RBAC3

RBACo - má nejméně požadavků, role zde nemají žádnou hierarchii či strukturu. Je to základní model, jsou zde základní entity: uživatel (users U), role (R) a povolení (permission P). Vztahy mezi nimi jsou dále modulovány pomocí omezení (constrains). Dále je zde množina sessions (s). Uživatel je lidský uživatel: je možno model upravit tak, aby se jednalo i o další uživatele, nejen lidi, ale zde je konkrétně myšlen člověk. Role je pracovní funkce nebo pracovní název v organizaci s nějakou přidruženou informací týkající se autority a odpovědnosti v této funkci. Povolení P je mód přístupu k jednomu nebo více objektům v systému. Jde o autorizaci, přístupová práva nebo povolení (různá literatura používá různé názvy). Povolení je vždy kladné a vyjadřuje schopnost držitele povolení provést nějakou akci na systému.

RBAC1 - množina rolí má hierarchickou strukturu, nadřazená role dědí povolené aktivity od podřízených rolí. Protože uspořádání nemusí být lineární, ale jde o částečné uspoř-



Obrázek 2.1: Vztah mezi RBAC modely

dání, může nadřízená role dědit více možností (např. vedoucí katedry dědí práva vyučujícího Programování i práva vyučujícího Databáze, vedoucí studijního oddělení dědí práva referentek pro všechny ročníky atd.)

RBAC2 používá omezení, která určují, zda různé varianty jednotlivých komponent modelů jsou povoleny (např. separace povinností, mohou být různé podmnožiny rolí, kde uživatel může být nucen nebýt členem více než jedné role).

RBAC3 kombinuje předchozí dvě a tím pádem i RBACo a dále přidává určitá omezení.

Modelům 1, 2, 3 se říká pokročilé modely. RBAC model formalizovali Ferraiolo a Kuhn. [7]. Ti kromě subjektu a objektu ještě zavádějí tzv. Transakce. Ve své práci zavedli aktivní roli subjektu AR (s:subjekt) - role, kterou subjekt právě vykonává. Autorizovanou roli subjektu RA (s: subjekt) - každý subjekt může být autorizován pro jednu či více rolí. TA (r:role) - transakce autorizovaná pro roli, každá role může být autorizována pro vykonávání jedné či více transakcí. Subjekt může vykonávat transakci, exec(s,t) je pravda, jestliže subjekt s může vykonávat transakci t současné době, jinak to je nepravda. [6]

2.2 Situace na poli wiki systémů

2.2.1 Historie wiki

22.března 2001 členové Seattle Wireless skupiny oznámili, že hlavní stránka jejích internetových stránek byla poškozena. Zdá se, že nekdo objevil odkaz, který umožňoval editaci jejich stránek a poškodil je. Nicméně správce měl uložené dřívější verze stránek a mohl je tak velmi snadno obnovit. Po tomto incidentu se zdálo, že stránky nemohou být sami od sebe editovatelné pro širokou veřejnost. Ovšem jak se ukázalo později na příkladu Wikipedie nebo C2.com, které byly založeny na podobných technologiích, možné to je. Tato technologie je nazývána jako WikiWikiWeb nebo WikiWiki nebo Wiki, a její hlavní zásadou je, že každá stránka webu může být editována kýmkoli kdykoli. Pro WWW stránky to byla radikální změna ve způsobu, jakým se zveřejňovaly informace, Wiki totiž znamená totální anarchii. Provoz prvních wiki stránek byl zahájen již v roce 1994. Termín wiki vytvořil Ward Cunningham, který je uznáván jako vynálezce tohoto pojmu.

Jedním z kritických aspektů vývoje softwearu je dokumentace. A to nejen hotová uživatelská příručka v době, kdy je program hotov, ale také ukládání technických specifikací

pro využití vývojáři během projektu. Tato technická dokumetnace se v průběhu času může měnit a může ji editovat mnoho lidí. Tradičně to bylo řešeno tak, že se dokumenty ukládali na sdíleném souborovém serveru. To ale mělo mnoho nevýhod. Co Ward udělal bylo, že navrhl malý CGI skript, který dovolil kterémukoli členovi skupiny aby okamžitě online aktualizoval dokumentaci. Každá změna byla zaznamenána a mohlo být přezkoumáno, jestli se jedná o užitečné informace. To se ukázalo jako velmi dobrý a rychlý způsob jak vytvořit a dodržovat technickou dokumentaci a Ward se rozhodl, že tento způsob pojmenuje wiki-wiki. Jedná se o hawajské slovo, které znamená rychlý. Později si otevřel veřejnou wiki o jeho společnosti C2.com, pro diskuze týkající se softwarového inženýrství a metodiky. Nazval ji Portland Pattern Repository. Na začátku roku 2001 to byla největší wiki na světě. Od té doby je wiki chápána jako nástroj budování komunit pro různé technologicky orientované zájmové skupiny.

Mezi méně úspěšné wiki online projekty patři Nupedia. Její majitel, Jimmy Wales, najal Larryho Sangera jako šéfredaktora Nupedie, aby koordinoval dobrovolné příspěvky a články a hodnotil je. Avšak mnoho příspěvků nepřibývalo a ukázalo se, že by trvalo dlouho, než by se to mohlo změnit a Nupedia by se mohla stát zdrojem vědomostí. A tak Sanger v lednu roku 2001 vytvořil paralelní projekt - Wikipedii - kde mohl editovat články každý uživatel. Wikipedie znamená začátek nové éry v použití wiki jako nástroje pro budování svobodné a otevřené encyklopedie. Tato wiki používala sofrware UseModWiki napsaný Cliffordem Adamsem. V prvním roce své existence se Wikipedie stala zdaleka největší wiki stránkou s více než 30 000 stranami a byl to tak první úspěšný pokus o vytvořený online encyklopedie. Wikipedie začala jako stránka v anglickém jazyce, ale její obrovský úspěch vedl k vytvořeni verzí v jiných jazycích.

2.2.2 Wiki technologie

Wiki je realizován jako součást webové stránky skriptem CGI nebo jinou podobnou skriptovací technologii. Wiki stránka sama o sobě je uložena buď v souboru nebo v databázi. Když prohlížeč vyžádá stránku, wiki skript přeloží čistý text do html kódu, který se stane součástí vracející se webové stránky. Kromě toho obsahuje stránka také hlavičky obsahující název stránky, navigační menu a odkazy. Nejdůležitější z odkazů - a to je podstata wiki konceptu - je „editovat stránku“. Kliknutím na odkaz „Upravit“ se přenese stejná stránka znovu, ale tentokrát se čistý HTML kód nepřekládá, ale je vytištěn ve velkém poli, kde může uživatel okamžitě stránku editovat a po uložení se změny okamžitě projeví na vzhledu stránky. Po potvrzení o uložení stránky jsou nová data zpracována wiki skriptem a za pomoci revizního systému jsou nová data uložena. Dřívější verze a rozdíly mezi nimy jsou k dispozici pomocí odkazů na wiki stránkách. Wiki stránky jsou psány ve formě běžného textu a do HTML kódu jsou překládany automaticky. Uživatel je tak oprostěn od učení se dalšího jazyku.

2.2.3 Řízení uživatelů

Většina veřejných wiki se vyhýbá povinným registračním procedurám. Přesto mnoho velkých wiki systémů (včetně MediaWiki, MoinMoin, UseModWiki a TWiki) poskytují způsob omezení přístupu pro zápis. Některé wiki umožňují zakázat editaci jednotlivým uživatelům blokováním jejich IP adresy nebo uživatelského jména. Protože internetoví poskytovatelé používají dynamicky přidělované adresy, může být blokování IP adresy snadno obcházeno. Proto je někdy používáno dočasné blokování celého intervalu IP adres, aby byla jistota, že vandal nemůže po nějaký čas editovat stránky. Předpokládá se, že ho to dostatečně od-

radí. Ale může to také omezit jiného neproblémového uživatele, připojeného přes stejného poskytovatele.

Obecná obrana před trvalým vandalem je jednoduše nechat ho znehodnotit kolik stránek chce s vědomím, že mohou být jednoduše sledovány a obnoveny, až toho vandal zanechá. Tato politika se může stát rychle nepraktická ve střetu se systematickým ničením, které nepochází ze vzteku nebo frustrace.

Jako bezpečnostní opatření mají některé wiki možnost přepnout databázi do read-only módu, zatímco jiné nasazují politiku, při které mohou pouze zavedení uživatelé až do určené doby pokračovat v editaci. Ale obecně řečeno, jakékoli škody uložené vandaly mohou být snadno a rychle napraveny. Problematičtější jsou malé chyby vložené do stránek, které projdou nezachycené. [4]

2.2.4 Sociální a právní aspekty

Většina lidí, když se dozvěděla na jakém principu wiki stránky fungují předpokládala, že tyto se stanou terčem útočníků a lidí, kteří využijí lákavé nabídky zničit něčí práci. To se ale nestalo. Je to způsobeno nejpíš tím, že wiki stránky jsou drženy pod správou verzí a tak je velmi jednoduché napravit škody, které někdo relativně pracně napáchal. Další problém mohl nastat v případě tzv. editačních válek. Ty nastanou v případě, že na jedno téma existuje více názorů. Takové války ale spotřebují mnoho energie a hlavně času a tak se wiki uživatelé naučili přijmout cizí názory a nebo neutrální postoj. Dobrý článek nebo příspěvek je ten, který zůstane nikým nezměněn, je objektivní a kompletní.

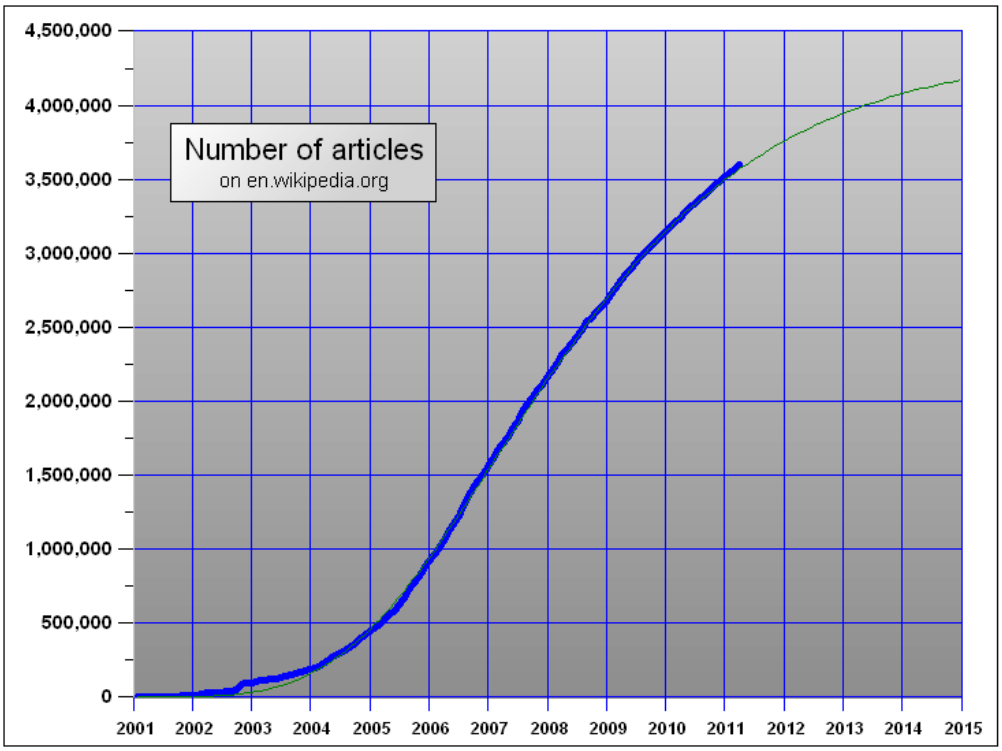
Zajímavější je ale možná otázka týkající se autorských práv. Poté, co několik lidí napíše článek nebo příspěvek na wiki, mají už nárok na autorská práva ke svému příspěvku? V interních wiki je obvykle výsledek dokumentace k projektu ve vlastnictví zaměstnavatele. Ale pro dobrovolné, open source projekty a wiki má každý přispěvatel silnější roli (větší zodpovědnost). Další problém s autorskými právy se týká použití materiálu chráněného autorskými právy na wiki. Pokud přispěvatel přidá na wiki článek, který je kryt něčími autorskými právy, kdo je odpovědný za možné vzniklé škody? Ve většině zemí je rozdíl v legislativě mezi klasickými tištěnými novinami, kde je zodpovědný za obsah vydavatel a webhostingovými službami a wiki systémy, kde je zodpovědný každý uživatel za svůj příspěvek. S ohledem na tyto právní otázky zaujala Wikipedie preventivní postoj. Každý přispěvatel poskytuje autorská práva ke svému příspěvku. Zároveň jsou informováni o tom, že po stisku tlačítka „Uložit“ souhlasí s dohodou, že jejich obsah je dostupný pro GNU Free Documentation License (FDL). Tento licenční text byl napsán Free Software Foundation (FSF) pro účely poskytnutí technické dokumentace. V podstatě to znamená, že kdokoli může volně kopírovat a používat text za předpokladu, že umožní dalším uživatelům přístup k tomuto výtvoru. Jeden ze spůsobů je zmínit URL wiki stránky, kde byl použitý obsah vyhledán.[5]

Z výše uvedených problémů by se mohlo zdát, že wiki trápí mnoho problémů, avšak tomu tak není. Každodenní život na wiki je plný radosti z učení a sbírka článků a příspěvků neustále roste. Všechny jsou úhledně upraveny a propojeny přes hypertextové odkazy.

2.2.5 Statistiky

Pro měření růstu a velikosti wiki byly vyvinuty různé metriky. Nejzákladnější parametr je počet stran a počet příspěvků za jednotku času. Jako nový příspěvek je počítáno každé stisknutí tlačítka „Uložit“. Tento postup ale nerozlišuje a nebere v úvahu velikost příspěvků. Protože je rozdíl mezi malou opravou chyby a novým velkým článkem. Jedním z řešení by

bylo počítat např. jenom příspěvky delší než 200 znaků a nebo 15 slov. Nicméně se uchytilo až jiné počítání. V průběhu prvního roku Wikipedie se začalo s tzv. počítáním čárek. Je to počet stránek s alespoň jednou čárkou. U Wikipedie to zahrnuje více než 80 % všech článků. Na začátku roku 2011 měla anglická verze wikipedie 3 621 603 obsahových stránek. Česká verze měla 193 626 stránek.



Obrázek 2.2: Nárust stránek na Wikipedii [3]

2.3 DokuWiki

2.3.1 Úvod do DokuWiki

Představme si, že pracujeme na nějakém projektu společne s kolegy z celého světa. Potřebujeme rychle nějaký online prostor, kde bysme mohli diskutovat a sdílet informace s ostatními (dokumentaci, apod.). Jak bylo psáno v předchozích odstavcích, na takové věci se nejlépe hodí wiki. Pokud nemáme mnoho času, potřebujeme rychlé, jednoduché a funkční řešení. A to je přesně to, co nám software DokuWiki nabízí. Jedná se o wiki napsanou v PHP, hodí se tedy na většinu hostingů. Navíc ani nepotřebuje databázi, všechna data totiž ukládá do textových souborů. Příjemné je, že DokuWiki si dobře poradí s obsahem v mnoha jazycích a navíc s námi komunikuje česky. Přístup ke své wiki lze jasně definovat pomocí pravidel (ACL). Můžeme například povolit zápis nebo celý přístup k určitým stránkám pouze pro registrované uživatele. DokuWiki v základní instalaci nabízí mnoho, ale navzdory tomu můžeme občas potřebovat některé další maličkosti. Na stránkách projektu lze najít mnoho zásuvných modulů a drobných modifikací.

2.3.2 Tvorba zásuvných modulů

V současné době nabízí DokuWiki 5 různých typů zásuvných modulů:

- Zásuvný modul rozšiřující syntaxy,
- Zásuvný modul Action,
- Zásuvný modul Admin,
- Zásuvný modul Helper,
- Zásuvný modul Renderer.

V závislosti na složitosti zásuvného modulu může obsahovat jeden nebo více typů. JavaScript a CSS styly mohou být použity u všech typů zásuvných modulů.

Zásuvný modul Action

Akční zásuvné moduly jsou navrženy pro práci s DokuWiki akcemi, s cílem umožnit rozšíření jakékoli části wiki, která je signalizována právě touto akcí. Akční zásuvné moduly jsou načteny před každým procesem (např. načtení stránky). Ihned po načtení každého akčního modulu je volána jeho metoda `register()`, která zásuvný modul zaregistruje ke zpracování události (event). Když je událost signalizována, volají se postupně všechny zásuvné moduly, které reagují na danou událost (bez zvláštního pořadí se řadí do zásobníku). Existují vlastně dva zásobníky: jeden pro „před“ DokuWiki akci a jeden pro „po“ DokuWiki akci. Ty jsou určené klíčovými slovy `BEFORE` a `AFTER` při registraci na akci.

Před zpracováním vlastní akce se se zpracují všechny `BEFORE` požadavky. To dává možnost uživateli změnit chování nebo vzhled stránky wiki dřív, než se dostane ke zpracování systémem. V případě událostí `IO_WIKIPAGE_WRITE` můžeme např. provést změny v obsahu stránky dřív, než se dostane k prohlížeči.

Po provedení vlastní akce se přechází ke zpracování požadavků, které jsou registrovány klíčovým slovem `AFTER`. V případě např. akce `TPL_ACT_RENDER` je možné připojit obsah na konec DokuWiki stránky. Ve zjednodušeném pseudokódu by mohl předchozí postup vypadat takto:

```
var $process_event = true;
var $default_stopped = false;

loop_through_BEFORE_List() {
    return if $process_event == false;
}

do_DokuWiki_Action() {
    return if $default_stopped;
}

var $process_event = true;

loop_through_AFTER_List(){
 return if $process_event == false;
}
```

Zásuvný modul Admin

Admin zásuvné moduly jsou zásuvné moduly, které poskytují extra funkce dostupné přes administrátorské okno. Uživatel spustí zásuvný modul kliknutím na odkaz v již zmiňovaném administrátorském menu. Tím se zavolá funkce `handle()`, která zpracovává dotazy a jako výstup slouží funkce `html()`, která má za úkol zajistit rozhraní mezi uživatelem a systémem. Tyto dvě funkce jsou zároveň jediné povinné. Následné už jsou dobrovolné a nemusí být implementovány:

- forAdminOnly - určuje, zda je zásuvný modul přístupný pouze pro uživatele administrátory,
- getMenuText - vrátí řetezec, kterým bude zásuvný modul rozpoznatelný v administrátorském menu,
- getMenuSort - vrátí řetezec, který určuje pořadí zásuvného modulu v administrátorském menu,
- getTOC - pomocí této funkce můžete vrátit obsah potenciálně dlouhých stránek.

Admin zásuvný modul komunikuje s uživatelem prostřednictvím DokuWiki dat vrácených pomocí formulářů a nebo pomocí dotazu na řetězce (tj. $ _REQUEST, $ _POST nebo $ _GET proměnné). Konkrétně to funguje tak, že ve funkci `html()` se nastavují příslušné formuláře a ty se posléze (po odeslání) zpracovávají ve funkci `handle()`.

Zásuvný modul Syntax

Syntax zásuvné moduly jsou moduly, které rozšiřují DokuWiki syntaxi.

Zásuvný modul Helper

Hepler zásuvné moduly usnadňují vývojářům zásuvných modulů využívat funkce jiných, stávajících zásuvných modulů.

Zásuvný modul Renderer

Render zásuvné moduly umožňují provádět různé mechanismy pro vykreslování DokuWiki stránek. DokuWiki analyzuje syntaxi do pole instrukcí, které se následně převádí na výstupní formát. Tento výstupní formát je obvykle XHTML. Zásuvné renderer modely umožňují autorům psát své vlastní vykreslovací výstupy v takovém formátu, jakém chtějí.

Platný název zásuvného modulu musí obsahovat pouze znaky a-z a 0-9. Nesmí obsahovat podtržítko. Název musí být unikátní, tzn. dva rozdílné moduly nesmí mít stejný název. Pokud autor vytvoří nový zásuvný modul, může se s ním podělit s ostatními tím, že ho zpřístupní na DokuWiki stránky. Přesněji musí založit novou wiki stránku ve jmenném prostoru `plugin`, která má stejný název, jako nový zásuvný modul. Např. Pokud je název nového zásuvného modulu `autlogin`, musí být vytvořena stránka `plugin:autlogin`. Stránka by měla obsahovat všechny potřebné informace, jak nainstalovat a používat zásuvný modul a dát uživatelům dobrou představu, co daný produkt dělá. Na dokuwiki.org není povolen upload, takže autor si musí zařídit svůj vlastní úložný prostor, přes který si daný zásuvný modul ostatní stáhnou. K 5.10.2010 bylo na stránkách DokuWiki zpřístupňeno 672 zásuvných modulů.

Kapitola 3

Návrh zásuvného modulu pro jednoduché přidělování práv

3.1 ACL zásuvný modul

Jako většina Wiki je v základu velmi otevřená. Kdokoliv může vytvářet, editovat a mazat stránky. Avšak někdy má význam omezit přístup na některé nebo všechny stránky. A tady vstupuje do hry Access Control Lists (seznam přístupových práv) (ACL). [8]

Pro zapnutí ACL musíme nejprve nastavit konfigurační volby. Do souboru `local.php` přidáme následující řádky:

```
$conf['useacl']       = 1;
$conf['superuser']    = '@admin';
```

Useacl povolí ACL. Jakmile je funkce puštěna, objeví se na konci každé wiki stránky přihlašovací tlačítko, a uživatel se může sám zaregistrovat. Volba superuser určuje kdo je v DokuWiki oprávněný k čemukoliv (včetně přidávání uživatelů) - tento může být buďto jednotlivý uživatel nebo skupina (označená uvozujícím @). Nyní můžeme nastavovat pravidla. Ta se buďto určují přes acl rozhraní v DokuWiki a nebo přímo psaním pravidel do souboru `conf/acl.user.php` (nedoporučuje se). Pravidla mají následující tvar:

```
*                   bigboss      16
start               @ALL          1
marketing:*         @marketing    8
devel:*             @ALL          0
devel:*             @devel        8
```

Na první pozici je stránka nebo jmenný prostor, kterého se pravidlo týká. Jmenný prostor je určen znakem „*“ na konci. Jako druhý je uživatel nebo skupina (skupina začíná znakem „@“). Jako poslední je číslo pravidla. Čím vyšší je toto pravidlo, tím větší oprávnění uživatel získává. Máme sedm úrovní oprávnění vyjádřených číslem (typu integer). Vysoká úroven obsahuje i nižší prvky. Zkrátka pokud můžeme upravovat stráknu, tak jí můžeme i číst. Admin s právy 255 by nikdy neměl být v souboru `conf/acl.auth.php`. Ten je použit jen interně v porovnání s volbou superuser. V souboru na pořadí nezáleží. Soubor je brán celý, takže se hledá dokonalá kombinace shody mezi stránkou a uživatelem. Když je nalezena shoda tak je další hledání ukončeno. Pokud nebude nalezena shoda, skupina práv

Jméno	Úroveň	Použitelnost	Práva	DokuWiki konstanta
none	0	stránky, jm. prostor	žádná práva	AUTH_NONE
read	1	stránky, jm. prostor	čtení	AUTH_READ
edit	2	stránky, jm. prostor	upravovat existující stránky	AUTH_EDIT
create	4	jm. prostor	vytvoření stránky	AUTH_CREATE
upload	8	jm. prostor	nahrání souborů	AUTH_UPLOAD
delete	16	jm. prostor	přepsání nebo smazání souborů	AUTH_DELETE
admin	255	jm. prostor	superuživatel(mění nastavení)	AUTH_ADMIN

Tabulka 3.1: Tabulka oprávnění

pro aktuální stránku je dál kontrolována. Pokud nebude nalezena shoda do konce souboru, tak kontrola pokračuje na dalším vyšším jmenném prostoru.

Nyní si podrobněji popíšeme, jak je programově řešeno nastavování ACL pravidel. Callgraf funkcí můžete nalézl v příloze A.1. Následuje popis činnosti funkcí.

handle() - zpracovává „dotazy“, které byly vytvořeny v html sekci

- _acl_init - načte aktuální nastavení ACL pravidel do vícerozměrného pole
- _acl_add - přidá nové ACL pravidlo do souboru `conf/acl.auth.php`
- _acl_del - vymaže pravidlo ze souboru `conf/acl.auth.php`

html() - tvoří rozhraní mezi uživatelem a systémem. Z funkce html jsou volány další funkce a tím se zajišťuje propojení mezi uživatelem a samotným systémem. V html je zapsán vzhled systému.

- `_html_explorer` - zobrazuje stromové menu k výběru stránky nebo jmenného prostoru, pro který chceme nastavit nebo editovat pravidlo
- `_html_buildlist` - sestaví neuspořádaný výčet z přiloženého pole. Používá funkce `_html_li_acl` a `_html_list_acl` a zobrazí samotnou stromovou strukturu menu.
- `get_tree` - načte strukturu stránek a jmenných prostorů, které jsou uloženy ve složkách a souborech pomocí proměnných `$conf['mediadir']` a `$conf['datadir']`
- `_html_detail` - zobrazuje Select (výběrové pole), nápovědu, editaci pravidel vybraného uživatele /skupiny
- `_html_select` - má za úkol zobrazit uživatele/skupinu pomocí výběrového pole (Select tag).
- `_html_info` - pokud pomocí předchozí funkce nebyl vybrán určitý uživatel nebo skupina, tak tiskne nápovědu. Jinak vytiskne editační nabídku s ACL pravidly, příslušící danému uživateli/skupině. Slouží k tomu funkce `_get_exact_perm` (jako vstupní proměnné slouží stránka/jmenný prostor a uživatel/skupina a vrátí ACL pravidla pro dané proměnné), `_html_explain`(vypíše slovně nastavená pravidla) a `_html_acleditor` (zobrazi ACL pravidla pomoci checkboxů).

Funkce `_html_explain`, `_get_exact_perm` a `_html_acleditor` se volají pouze v případě, byl-li vybrán ve funkci `_html_select` určitý uživatel/skupina. Pokud ne, tiskne se nápověda pomocí fce. `locale_xhtml('help.txt')` kde se vytiskne to, co je uloženo v souboru `text.php`.

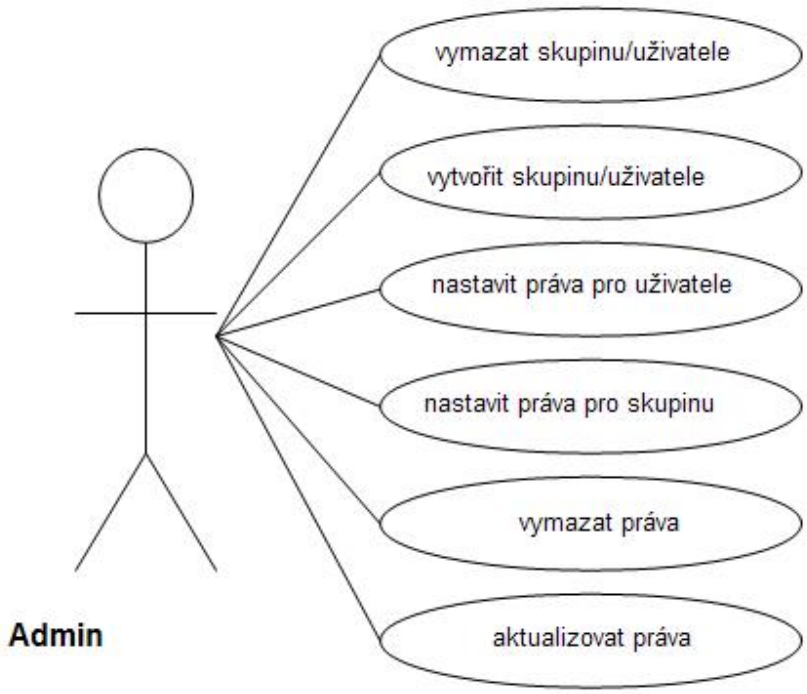
- `_html_table` - zobrazí všechna pravidla, která jsou aktuálně nastavena pro všechny uživatele. Zároveň je může admin mazat.

Signály:

- submit - po stisku tlačítka z funkcí `_html_detail`, `_html_table` nebo `_html_acleditor` se volá funkce handle, která vyhodnotí, které proměnné byly nastaveny (dotaz `$_REQUEST['....']`) a podle toho provede určitou operaci (uložení/vymazání, update pravidla,..).
- change - při změně výběru uživatele/skupiny pomocí Select výběrového pole je „zachycen“ signál a volána fce. `_html_info`, která vytiskne příslušné informace pro nově vybraný objekt.
- click - pokud klikne uživatel na stránku/jmenný prostor ve stromovém menu, je volána fce. `_html_info`, která vytiskne příslušné informace pro nově vybranou stránku.

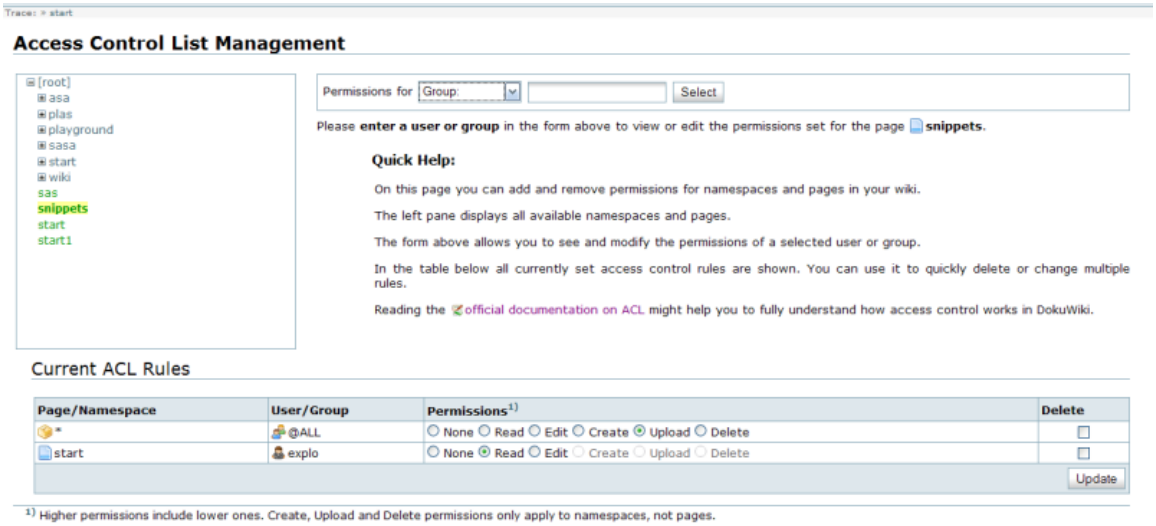
Tyto signály jsou realizovány pomocí JavaScriptu. JavaScript musí být schopen lokalizovat objekt, kterého se signál týká. Nejjednodušší způsob, jak najít tento objekt je poskytnout související HTML tag ID. Toto ID musí být unikátní mezi všemi ID na stránce.

`Diagram užití`



Obrázek 3.1: Diagram užití

Na obrázku 3.2 vidíme ukázku zadávání práv. V levé části je stromové menu, kde vybíráme stránku nebo jmenný prostor a pomocí select boxů vybíráme uživatele nebo skupinu, poté se objeví checkboxy, kde nastavíme právo a potvrdíme odesláním. Ve spodní části jsou pak v tabulce vypsána všechna současná pravidla.



Obrázek 3.2: Zadávání práv

Jak je z předchozího vidět, současná situace v DokuWiki nereflektuje žádná moderátorská práva. Tzn. práva, která by daného moderátora opravňovala spravovat určitou stránku nebo jmenný prostor a to tím způsobem, že by byl oproti administrátorovi zbaven práva vytvářet a mazat uživatele.

3.2 Navrhnutá vylepšení

Největší nevýhodou u předchozího přidělování práv jsou právě moderátorská práva.

Moderátor

Moderátor má vyšší oprávnění než běžný uživatel, který nemá žádná práva pro správu stránek. Moderátor by měl mít možnost:

- přidělování práv
- aktualizace práv
- mazání práv

Oproti administrátorovi uživateli, který má nejvyšší oprávnění je moderátor omezen v možnostech:

- vytváření uživatelů a skupin
- mazání uživatelů a skupin

Udělit oprávnění moderovat stránky musí mít pouze administrátor. Ten po vybrání stránky, nebo jmenného prostoru pomocí stromového menu musí vybrat uživatele, kterému chce přidělit moderátorský status. Po vybrání těchto povinných informací stiskne klávesu pro uložení. Informace o tom, kdo je moderátor a jaké stránky (jmenné prostory) má právo spravovat se ukládá do souboru. V tomto souboru by se informace ukládaly následovným způsobem:

```
uživatel       stránka/jm._prostor
```

Takže v konkrétním případě by situace mohla vypadat následovně:

```
petr    start
admin    *
explo    wiki:*
```

Nyní máme určeno a uloženo kdo se stal moderátorem a musíme vytvořit moderátorskou stránku. Po přihlášení uživatele zkontrolujeme, zda-li jsou mu nastavena práva moderovat. To zjistíme tak, že načteme soubor, kde jsou uloženi všichni moderátoři a pokud je v něm uložen i přihlášený uživatel, je moderátorem. Pro přístup k moderátorské stránce použijeme tlačítko, které se objeví pouze uživatelům, kteří mají moderátorská práva. Po kliknutí na toto tlačítko je načtena moderátorská stránka. Ta je složena ze dvou částí:

1. možnost zadávání nových práv
2. všechna práva vypsaná do tabulky

Pro obě části můžeme použít již existujících funkcí. Jestliže bychom chtěli zadat a uložit nové právo, nejprve musíme vybrat stránku nebo jmenný prostor, následně uživatele nebo skupinu, které chceme nové oprávnění udělit a jako poslední vybereme pomocí checkboxů maximální právo. Po odeslání formuláře můžeme využít funkce `_acl_add`, která uloží nové pravidlo do souboru `conf/acl.auth.php`. Pro výpis pravidel můžeme použít funkci `_html_table`. Jediný rozdíl oproti této funkci použité pro výpis pravidel v administrátorském menu je ten, že nechceme vypsat všechna pravidla, ale pouze ta, která se týkají stránek nebo jmenných prostorů, které daný uživatel má právo moderovat. Toho dosáhneme tím, že jako vstupní parametr této funkce je seznam stránek a jmenných prostorů, pro které mají být vypsána pravidla. V administrátorském menu to budou všechny stránky. U moderátorské stránky musíme nejprve zjistit, které stránky má moderátor právo spravovat a ty potom funkci předložíme jako vstupní parametr.

Kapitola 4

Koncept uživatele hosta a jeho automatické přihlašování

4.1 Autentizace a autorizace

Autorizace

Proces autorizace označuje získání přístupu k informacím, funkcím a dalším objektům, který se skládá z:

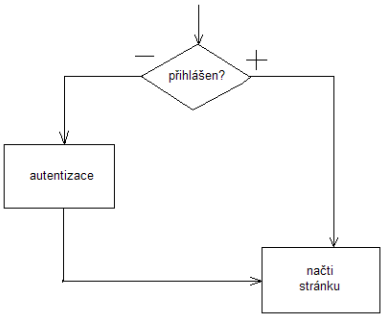
- autentizace subjektu (zjištění jeho identity)
- vyhledání v seznamu oprávněných subjektů, jejich rolí a práv
- udělení oprávnění nebo odepření přístupu

Seznam oprávnění je v informatice realizován přidělením oprávnění na soubory, adresáře, provedení operace nebo přístupu k jiným prostředkům v počítači. Autorizaci provádí operační systém nebo specializovaný software na základě seznamů pro řízení přístupu. [2] V našem případě má autorizaci na starosti administrátorská část zásuvného modulu.

Autentizace

Je proces ověření proklamované identity subjektu. Proběhne-li proces autentizace, dojde k autorizaci. Autentizace je v informatice ověření identity uživatele služeb nebo původce zprávy. Používají se tyto základní metody pro zjištění identity:

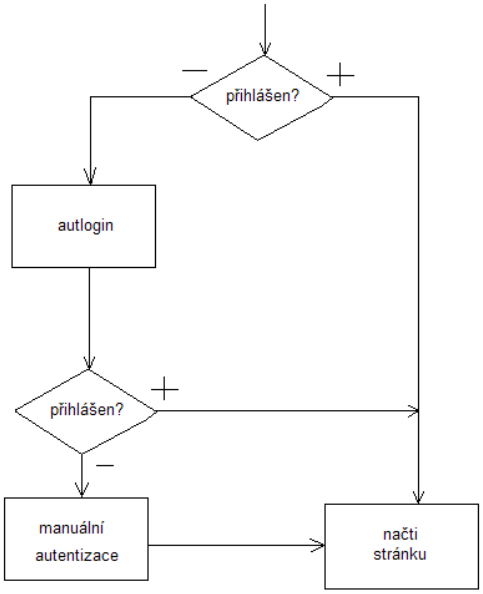
- podle toho, co uživatel zná (zná správnou kombinaci uživatelského označení a hesla nebo PIN)
- podle toho, co uživatel má (nějaký technický prostředek, který uživatel vlastní - hardwarový klíč, smart card, privátní klíč apod.)
- podle toho, čím uživatel je (uživatel má biometrické vlastnosti, které lze prověřit - otisk prstu, snímek oční duhovky či sítnice apod.)
- podle toho, co uživatel umí (umí správně odpovědět na náhodně vygenerovaný kontrolní dotaz)



Obrázek 4.1: Systém autentizace

[1] V našem případě se o autorizaci stará akční část našeho zásuvného modulu.

Jak je vidět na obrázku 4.1, tak současný systém nenabízí žádnou možnost automatického přihlášení, ale pouze ručního (manuálního) přihlášení.



Obrázek 4.2: Systém autentizace + autlogin

Automatické přihlašování zastává funkci autentizace a aplikuje se pouze na hosty. Tzn. na uživatele, kteří nejsou ještě přihlášeni. Pokud se automaticky uplatní pravidlo na hosta a je přihlášen, je načtena stránka. Pokud však pro něj neplatí žádné pravidlo, uživatel má možnost manuálního přihlášení (viz. 4.2).

Tento zásuvný modul se skládá ze dvou typů zásuvných modulů (byly popsány v kapitole 2.3.2). První část je Action zásuvný modul, a druhý Admin zásuvný modul.

4.2 Administrátorská část

Na úvod si uvedeme soubory, s kterými při vytváření zásuvného modulu pracujeme. Vyjasníme si, co do jednotlivých souborů ukládáme za informace.

visit.php

Do tohoto souboru ukládáme informace o nově příchozích hostech, pro které neplatí žádné pravidlo pro automatické přihlášeni. Ukládáme do něj informace ve tvaru:
`datum navštívená_stránka kritéria`
Pro lepší pochopení uvedu příklad:

```
03/05/2011-15:42:26 start IP=217.115.249.198,WB=Firefox,OS=Windows_XP
```

transl.php

Tento soubor obsahuje jméno uživatele, kritéria, kterými je určen, stránku, pro kterou je dané pravidlo platné a číslo pravidla. Tento soubor prakticky zastává stejnou funkci jako systémový soubor `conf/acl.auth.php` s tím rozdílem, že v našem souboru se navíc ukládají zmíněná kritéria. Informace jsou uloženy ve tvaru:
`stránka/jmenný_prostor kritéria jméno_uživatele číslo_práva`
Příklad záznamu ze souboru:

```
asa:* WB=Mozilla,VER=8.0,OS=Linux john 1
```

moderators.php

Tento soubor uchovává informace o moderátorech a o stránkách, které mohou daní uživatelé moderovat.Informace jsou uloženy ve tvaru:
`jméno_uživatele stránka/jmenný_prostor`
Příklad záznamu ze souboru:

```
admin  start:*
```

Ve zbylých dvou soborech jsou uloženy informace jaké operační systémy a jaké prohlížeče je schopen systém rozeznávat(systems.php - systémy, browsers.php - prohlížeče).

Administrátorská část je přístupná pouze pro uživatele, kteří mají status administrátora. Ale to nemusí být vždy pravda. Přístup můžeme umožnit i uživatelům, kteří patří do námi definované skupiny. Ta se nastaví v administrátorském menu a přes odkaz *správa nastavení* nastavíme řádek *„Manažer - skupina nebo uživatel s přístupem k některým správcovským funkcím“* na skupinu, kterou chceme určit jako manažerskou. Po tomto nastavení ještě musíme umožnit přístup této skupině do administrátorské části zásuvného modulu (nastavením funce `forAdminOnly` na false). Toto nastavení nám zaručí, že budou mít přístup

k moderátorské stránce i uživatelé, kteří byli nastaveni jako moderátoři, ale zároveň nejsou administrátory.

Jak již bylo řečeno v předchozích kapitolách 2.3.2, tak v případě Admin zásuvného modulu jsou povinné pouze 2 funkce.

4.2.1 html()

Tato fukce je pomyslně rozdělena na 2 části:

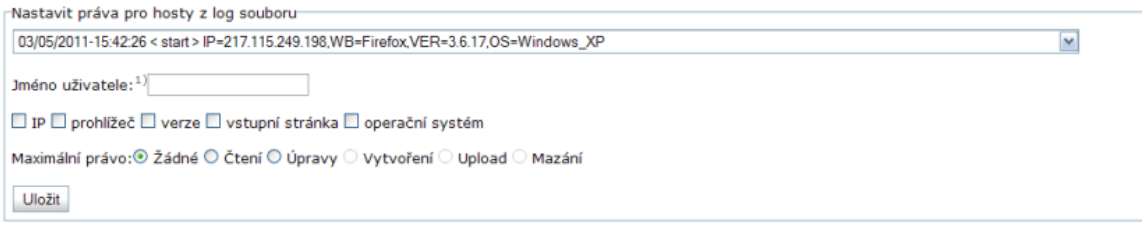
- část určena pouze pro administrátory
- část určená pro moderátory

Toto rozdělení je realizováno pomocí knihovní funkce `auth_isadmin()` . Tato nám vrací, pokud se jedná o administrátora, hodnotu true, jinak false. Je-li zjištěno, že přihlášen je administrátor, je mu zpřístupněna část, která je určena pro zadávání práv. Tato část je ještě rozdělena na 4 oddíly:

1. nastavit práva pro hosty z log souboru
2. manuální zadání práv
3. všechna nastavená práva
4. nastavení moderátorů

Nastavení práv pro hosty z log souboru

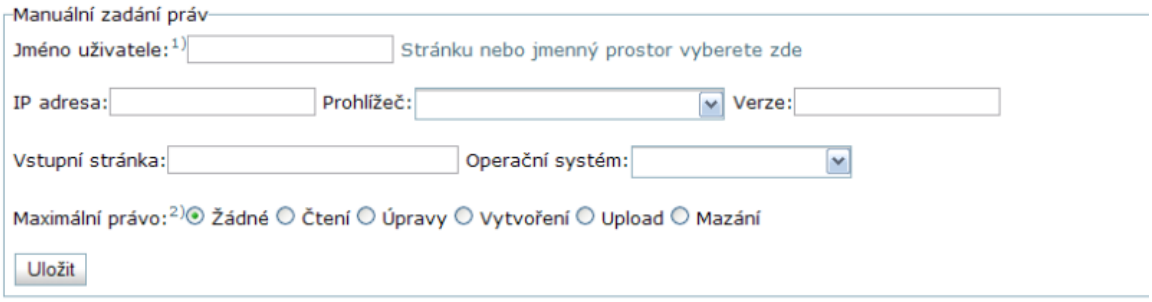
Každý, kdo navštíví naše wiki stránky, které spravujeme, a nesplňuje podmínky pro automatické přihlášení, je zapsán do souboru `settings/visit.php`. Tento soubor nám ukládá informace o nově příchozích hostech (IP adresa, prohlížeč, verze prohlížeče, operační systém a vstupní stránka). V administrátorské části zásuvného modulu je tento soubor využíván tak, že můžeme vybrat hosta zapsaného v tomto souboru, můžeme vybrat jaká kritéria chceme přiřadit k možnosti jeho automatického přihlášení a přidělit mu maximální právo. Takto vybraného uživatele pak uložíme a při jeho přístí návštěvě (pokud přistoupí ze stejného pc a prohlížeče) je tento host automaticky přihlášen. Konkrétně vybereme pomocí Select nabídky hosta, který je zapsán v log souboru, přiřadíme mu jméno, zaškrtneme alespoň jedno z kritérií, a jedno z práva (žádné, čtení, úpravy).



Obrázek 4.3: Zadávání práv pro hosty z log souboru

Manuální zadávání práv

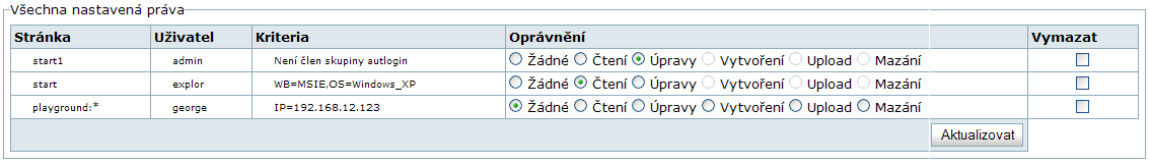
V této části nejsme svázání žádnými předurčenými údaji, ale můžeme je vyplňovat ručně. Stránku nebo jmenný prostor vybereme pomocí stromového menu. Protože stromové menu se používá pro výběr stránky ještě u zadávání moderátorů, není toto menu přímo u ostatních údajů, ale dostaneme se k němu přes odkaz, který nás nasměruje do horní části stránky, kde se toto menu nachází. Zadáme jméno uživatele a minimálně jedno z kritérií (IP adresa, prohlížeč,...). Typ prohlížeče a typ operačního systému vybíráme pomocí select nabídky, kde máme přednastaveno, které typy je systém schopen rozeznat. Nyní můžeme právo odeslat.



Obrázek 4.4: Manuální zadávání práv

Všechna nastavená práva

Tato část obsahuje tabulku, kde jsou vypsána všechna pravidla, která byla nastavena. Pokud bylo zadáno pravidlo v předchozích dvou případech (z log souboru nebo ručně) tak se v kolonce Kritéria zobrazí zadaná a vyplněná kritéria, pokud bylo právo nastaveno v klasickém a původním acl zásuvném modulu, je v této kolonce vypsáno „Není členem skupiny autlogin“. V této části může administrátor upravovat maximální právo a zároveň může jednotlivá práva mazat.



Obrázek 4.5: Tabulka práv

Nastavení moderátorů

Moderátora nastavíme tak, že vybereme jednoho existujcího uživatele, kteří jsou zobrazeni pomoci select nabídky. Poté jako v případě ručního zadávání práv jsme přes odkaz nasměrování na stromové menu, ve kterém vyberemu stránku nebo jmenný prostor, který chceme uživateli svěřit pod správu. Součástí této části je také tabulka, ve které jsou vypsáni všichni moderátoři a stránky nebo jmenné prostory, které spravují. Je zde také možnost moderátory mazat (viz. 4.6).

Obrázek 4.6: Zadávání moderátorů

4.2.2 handle()

Tato funkce je volána po odeslání html formuláře (form action=„“ method=„post“) z `html()` části zásuvného modulu. Jako první je testováno, zda-li byla ve stromovém menu vybrána konkrétní stránka nebo jmenný prostor. Pokud ano, tak je toto uloženo do proměnné `$ns`. Nyní se dostáváme k hlavní části této funkce. Zde je testováno, zda-li byl odeslán nějaký formulář (`$_REQUEST['cmd']`). Zároveň je pomocí funkce `checkSecurityToken()` testována bezpečnost. Tato funkce zkontroluje, jestli je posílaný bezpečnostní klíč platný. Tato funkce chrání proti padělání dotazů. Jestliže je vše v pořádku, tak se zjistí konkrétní dotaz, který byl odeslán.

Bylo zadáno pravidlo z log souboru

Je-li nastaven `$_REQUEST['cmd']['visit']` je jasné, že bylo zadáno pravidlo pro hosta, který je zapsán v log souboru. Postupně pomocí `$_REQUEST` zjistíme, která konkrétní kritéria byla vybrána. Dále uložíme do proměnné `$perm` číslo pravidla, které bylo zadáno (žádné, čtení,...). Jako poslední uložíme jméno, kterým jsme uživatele pojmenovali. Zadávání jména je nepovinný údaj. Pokud jméno nezadáme, je automaticky generováno funkcí `rand_alias()` (tato funkce nám vráti řetezec začínající jako „host“ a dalších 8 náhodných znaků a-z).

Nyní máme uloženy všechny potřebné proměnné a můžeme zavolat funkci `_save_user`. Vstupními parametry jsou: jaké pravidlo z log souboru bylo vybráno, číslo pravidla a kritéria. Hlavním úkolem této funkce je vyparsovat ze souboru `settings/visit.php` konkrétní kritéria. Tzn., že bylo-li vybráno jako jedno z kritérií IP adresa, tato funkce najde řádek s uživatelem, který má být uložen a najde jeho IP adresu. Následně je zjišteno, zda už neexistuje pravidlo pro uživatele se stejnými kritérii. Pokud ano, musí mu být přiřazeno stejné jméno, jako už v existujícím případě (nehledě na to, že administrátor zadal jiné jméno). Jestliže se zakládá nový uživatel, je mu vytvořen účet a je vytvořeno nové pravidlo do souboru `settings/transl.php`. Pokud už uživatel se stejnými kritérii existuje, je vytváření nového účtu přeskočeno a jenom je vytvořeno nové pravilo. Jako poslední musí následovat zapsání pravidla do konfiguračního DokuWiki souboru `conf/acl.auth.php`.

Aktualizace stávajícího pravidla

Jako další testujeme, zda administrátor neaktualizoval nebo nesmazal některé pravidlo. Pokud je nastaven `$_REQUEST['cmd']['update']` nastala právě tato situace. Nejprve projdeme dvojrozměrné pole, kde by mohla být uložena pravidla, která mají být vymazána. Toto pole je ve tvaru `stránka jméno`. Je-li toto pole naplněno informacemi o pravidlech, která mají být smazána, pro každé takové pravidlo se volá funkce `del_acl`, která vymaže

pravidlo ze souboru `conf/acl.auth.php`. Po smazání všech pravidel, kterÁ byl k zániku určena se tyto změny musí projevit i v souboru `settings/transl.php`. To nám obstará funkce `_actaulize()`. V dalším kroku postupně projdeme všechna pravidla a zjistíme, zda bylo u některého aktualizováno číslo pravidla. Pokud zjistíme, že se u některého takto stalo, je toto pravidlo nejprve vymazáno a poté uloženo s novým číslem pravidla.

Manuální zadání práva

Je signalizováno nastavením `$_REQUEST['cmd']['manual']` . Následně jsou uloženy do proměnných číslo pravidla, stránka nebo jmenný prostor, pro které je pravidlo nastavováno. Jako poslední se ukládají kritéria. Pokud byla zadána IP adresa, je kontrolováno, jestli je platná. To nám zajistí funkce `control_ip`. Pomocí regulárního výrazu zjistíme, zda je zadaná adresa platná IP adresa.Pokud ano vrací funkce pozitivní výsledek 1. Pokud ne, vrací -1. Po ověření IP adresy ukládáme postupně všechna ostatní kritéria, která byla vyplněna. Avšak u vstupní stránky, stejně jako u IP adresy voláme kontrolní funkci `control_page`. Tato funkce nám zjistí pomocí regulárního výrazu platost url adresy. Adresa musí začínat jedním z klíčových slov http,https nebo ftp. Je-li adresa v platném znění, je vrácen pozitivní výsledek. Je-li vyplněno alespoň jedno kritérium, volá se funkce `save`.

Tato funkce zjistí, jestli již existuje uživatel se stejnými kritérii. Pokud ano, přiřadí mu stejné jméno jako má již existující uživatel. Pokud není zjištěno, že daná kritéria již existují, je založen nový účet pro nového uživatele, a zapsáno nové pravidlo do souborů `conf/acl.auth.php` a `settings/transl.php`. Jestli existuje uživatel se stejnými kritérii, je pod stejným jménem zapsáno nové pravidlo taktéž do souborů `conf/acl.auth.php` a `settings/transl.php`.

Byl nastaven moderátor

Signalizuje aktivita `$_REQUEST['cmd']['moderator']` . Pokud je aktivní, je zjištěno, jestli pro danou stránku nebo jmenný prostor a daného uživatele již neexistuje stejný záznam v souboru `settings/moderators.php`. Pokud ne, je tento nový moderátor do tohoto souboru zapsán. Jetliže je to první záznam pro tohoto uživatele, musíme ho ještě zařadit do skupiny *„moderator“*. To udělá funkce `add_group()`. Tato funkce přidá do souboru `conf/users.auth.php` k našemu uživateli k jeho stávajícím skupinám zmiňovanou skupinu *„moderator“*.

Smazání moderátorského pravidla

Při aktivitě proměnné `$_REQUEST['cmd']['update_mod']` je zřejmé že bylo smazáno pravidlo. Všechna pravidla, která mají být smazána jsou uložena ve dvojrozměrném poli (jméno moderátora a stránka/jmenný prostor). Toto pole je postupně procházeno a tato práva pro moderátory jdou postupně ze souboru `settings/moderators.php` mazána. Po smazání každého pravidla je zjištěno, zda pro daného uživatele (moderátora), existuje v tomto souboru alespoň jeden záznam. Pokud ne, ztrácí tento uživatel status moderátora a musí být z jeho profilu odstraněna skupina *„moderator“*. To nám zajistí funkce `del_group`.

4.3 Akční část

Všechny akční zásuvné moduly jsou založeny na principu práce s událostmi (events) v DokuWiki systému. Tyto zásuvné moduly mohou rozšířit jakoukoli část DokuWiki, která je

signalizována pomocí události. Tento konkrétní zásuvný modul je založen na odchytávání událostí `TPL_ACT_RENDER` a `TPL_METAHEADER_OUTPUT`.

Po zjištění aktivity signálu `TPL_METAHEADER_OUTPUT` se jako první rozparsuje hlavička `$_SERVER` . Tato proměnná obsahuje důležité informace, se kterými budeme nadále pracovat. Jako první zjistíme, zda-li je nastavena hlavička `$_SERVER['REMOTE_USER']` . Tato obsahuje přihlašovací jméno přihlášeného uživatele. Pokud je tato proměnná nastavena na nějakou hodnotu, znamená to, že je již nějaký uživatel přihlášen a náš zásuvný modul končí svou činnost. Je to z toho důvodu, že zásuvný modul má za úkol automaticky přihlašovat hosty (nepřihlášené uživatele). Pokud je však zjištěno, že není nikdo přihlášen, prozkoumá se hlavička `$_SERVER['REMOTE_ADDR']` . Zde je umístěna IP adresa připojeného uživatele. Tato se uloží do proměnné. Stejně tak je testována hlavička `$_SERVER['HTTP_REFERER']` . Zde může být uložena vstupní stránka. Vstupní stránka je vlastně URL adresa, na které je umístěn odkaz na DokuWiki stránky. Je-li nastavena, tak stejně jako u IP adresy, je uložena do proměnné. Pokračuje se rozpársováním `$_SERVER['HTTP_USER_AGENT']` , kde je uložen typ a verze prohlížeče. Navíc je zde uložen i operační systém, který host používá. Zjištění těchto informací nám obstará funkce `detect_browser()`. Jako vstupní parametr je právě hlavička `$_SERVER['HTTP_USER_AGENT']` . Nejdříve je pomocí knihovní funkce `preg_match` zjištěn prohlížeč a jeho verze. Seznam všech prohlížečů, které je systém schopen detekovat je uložen v souboru `settings/browsers.php`. Následuje detekce operačního systému. Všechny druhy detekovaných operačních systémů jsou uloženy v souboru `settings/systems.php`. Pokud není možné detekovat některou z uvedených informací (např. verze prohlížeče), je příslušná proměnná nastavena na „unknown“. Po uzjištění těchto informací přecházíme na vyhledávání, zda-li splňuje náš host nějaké pravidlo.

Toto obstará funkce `find_best()`. Postupně se pomocí příkazu `foreach` prochází všechna pravidla ze souboru `settings/transl.php`. Na první pozici každého řádku v tomto souboru je stránka nebo jmenný prostor, kterého se toto pravidlo týká. Pomocí funcke `_is_page` zjistíme, zda-li je pravidlo určeno pro naši aktuální stránku (aktuální stránka je stejná jako stránka určená pro pravidlo a nebo aktuální stránka leží ve jmenném prostoru určeného pro pravidlo). Pokud ne, přejde se na nové pravidlo a celý proces porovnávání stránek se opakuje. Pokud však stránka souhlasí, načtou se kritéria a porovnávají se s údaji, které jsme zjistili o našem hostovi. Pokud host nesplňuje všechna kritéria u našeho pravidla, toto pravidlo se neuplatní a začíná se porovnáváním stránky na dalším pravidle. Pokud však náš uživatel splnil všechna předepsaná kritéria, je uloženo jméno, které náleží tomuto pravidlu. Ale tím naše hledání nekončí a musíme projít všechna pravidla do konce souboru. Může nastat totiž případ, že jsme našli shodu u nějakého pravidla, avšak dál v našem souboru `settings/transl.php` leží jiné pravidlo, které je specifičtější. Pro lepší pochopení uvedu příklad.

Naši wiki stránku načetl uživatel. Tento uživatel splňuje pravidlo:

```
start  IP=192.168.12.123  george 0
```

Avšak dále v souboru je pravidlo:

```
start IP=192.168.12.123,WB=MSIE,OS=Windows_XP john 1
```

Pokud náš uživatel splňuje všechna kritéria tohoto druhého pravidla i kritérium (IP adresu) v prvním pravidle, je přihlášen jako uživatel se jménem *john*, protože toto pravidlo je specifičtější pro našeho uživatele. Proto se vždy, když dané pravidlo splňuje všechny podmínky (kritéria) porovnává, kolik kritérií toto pravidlo obsahuje a pokud je toto číslo větší než zatím aktuálně největší, je toto nové pravidlo prohlášeno za momentálně nejlepší.
Po projití všech pravidel mohou nastat 2 situace:

1. žádné pravidlo není vhodné pro účel přihlášení
2. je vybráno pravidlo, jehož všechna kritéria uživatel splňuje

Pokud je vybráno pravidlo pro přihlášení, je náš host přihlášen pod jménem tohoto pravidla příkazem `auth_login()`. Pokud však žádné pravidlo vybráno nebylo, zapíše se návštěva hosta do souboru `settings/visit.php` pomocí funkce `write_to_visit()`. Tato funkce nejdřív zjistí, zda-li již není totožný záznam v tomto souboru uveden. Pokud ano, znova se nezapisuje. Pokud není nalezena shoda, je do souboru zapsán.

Druhým signálem, který se u tohoto zásuvného modulu odchytává je `TPL_ACT_RENDER`. Tento signál nám pomůže připojit na konec každé stránky (pokud je uživatel moderátor) tlačítko, pomocí něhož se uživatel dostane do moderátorského menu.

Po zjištění aktivity tohoto signálu je volána funkce `_ismoderator`, která zjistí, zda je přihlášený uživatel moderátor. Pokud ano, je zobrazeno tlačítko, které uživatele přesměruje na moderátorskou stránku, pokud ne, tlačítko zůstane skryto.

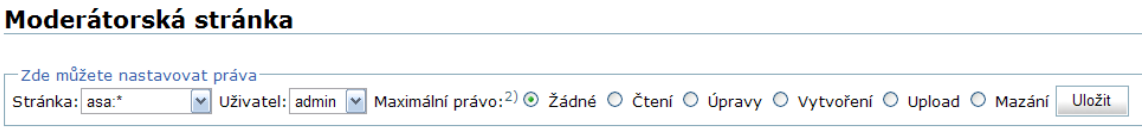
4.4 Moderátorká stránka

Moderátorská stránka je obsažena v administrátorské části zásuvného modulu. Ve funkci `html()` je část, která se aplikuje pouze pro uživatele moderátory. To zajistí podmínka `auth_ismanager()`. Moderátorská stránka se skládá ze dvou částí:

1. zadávání pravidel
2. všechna pravidla vypsaná do tabulky

Zadávání pravidel

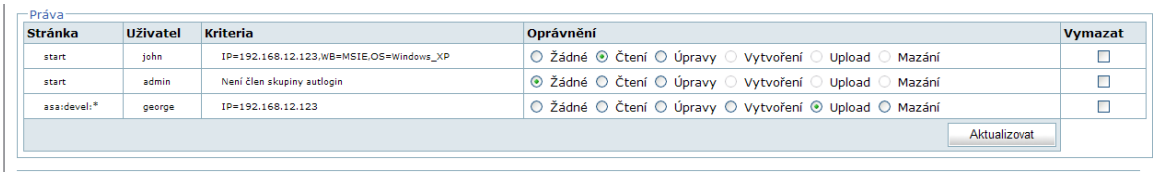
Tato část slouží moderátorovi k zadávání nových pravidel pro již existující uživatele. Omezení na jenom existující uživatele je z toho důvodu, že koncept moderátora je omezen tím, že nemůže samotné uživatele vytvářet. Stránku pro nové pravidlo vybereme pomocí select nabídky. V této nabídce nalezneme pouze stránky a jmenné prostory, na které má uživatel moderátorská práva. Po vybrání stránky vybereme uživatele taktéž pomocí select nabídky. Zvolíme mu pomocí checkboxů maximální pravidlo a uložíme.



Obrázek 4.7: Zadávání práv

Pravidla vypsaná do tabulky

Tato část obsahuje tabulku, kde jsou vypsána pravidla pro stránky, na které má uživatel moderátorské právo. Součástí této tabulky je možnost aktualizovat pravidla a popřípadě tato pravidla mazat.



Obrázek 4.8: Tabulka práv

V administrátorské části ve funkci `handle()` je obsažena funkčnost moderátorské stránky. Tato funkčnost je založena na testování, zda byl z `html()` odeslán formulář. Mohly být odeslány dva druhy formuláře:

1. bylo aktualizováno nebo smazáno stávající pravidlo
2. bylo nastaveno nové pravidlo

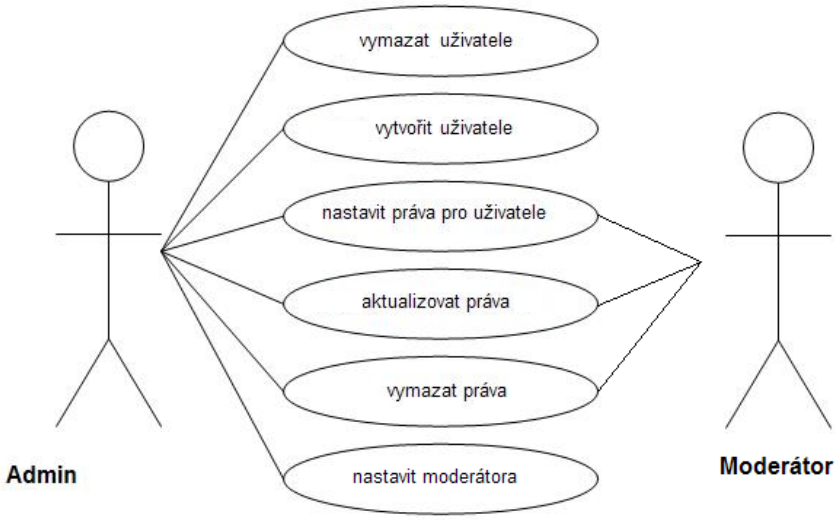
Bylo aktualizováno nebo smazáno stávající pravidlo

Tento případ pracuje na stejném principu jako Mazání a aktualizace pravidel v kapitole 4.2.2 (odstavec *Aktualizace stávajícího pravidla*).

Bylo nastaveno nové pravidlo

Po testu a zjištění aktivity `$_REQUEST['cmd']['set_mod']` zjistíme, zda-li již existuje stejné pravidlo. Pokud ne, uložíme zadané nové pravidlo do souborů `settings/transl.php` a `conf/acl.auth.php`.

Zavedli jsme tedy status moderátora a přiřadili mu úkoly, které má právo dělat. V následujícím diagramu jsou znázorněny všechny úkoly, které mohou administrátoři a moderátoři provádět.



Obrázek 4.9: Diagram užití

4.5 Úprava pomocí JavaScriptu

Na stromové menu, aby bylo dynamické, je použit JavaScript. Je obsahem souboru `script.js`. Javascript odchytává událost `click` v objektu `acl__tree`. Tento objekt je reprezentován HTML tagem s ID = „acl__tree“. Po odchycení události (kliknutím na stromové menu) je volána funkce `treehandler`. Ve funkci mohou nastat 2 případy:

1. Bylo kliknuto na obrázek - pokud jsme kliknuli na obrázek, chceme srolovat nebo rozvinou větev stromu. Pokud chceme větev srolovat, tak místo mínus obrázku vložíme k větvi obrázek plus a tuto větev srolujeme(schováme pododkazy). Pokud chceme větev rozvinout, vložíme obrázek mínus a přes soubor `ajax.php` tuto větev rozvineme.

2. Bylo kliknuto na odkaz - po označení stránky nebo jmenného prostoru nastavíme proměnnou `$current_ns` nebo `$current_id` na příslušnou hodnotu odkazu, na který jsme klikli.

Obsah souboru `script.js` je převzat a upraven z acl zásuvného modulu. Jelikož všechny zásuvné moduly a zdrojové kódy, ze kterých se skládají, spadají pod GPL (General Public License) mohli jsme ho použít za předpokladu, že dáme k dospizici ostatním uživatelům naše zdrojové kódy.

Kapitola 5

Závěr

Zásuvný modul, který byl základem k vytvoření této bakalářské práce vychází z již existujícího ACL zásuvného modulu. Rozšiřuje jeho možnosti a zároveň ho vylepšuje o vlastnosti, kdy se přihlašovat nemusí jednotliví uživatelé, ale může administrátor volit různé kombinace kritérií potřebných k automatickému přihlášení. Toho mohou využít správci wiki stránek, kteří chtějí mít větší přehled o tom, kdo navštěvuje jejich stránky a mohou tak lépe kontrolovat a omezovat přístup pro ostatní uživatele. Tento zásuvný modul nebudou zřejmě využívat uživatelé obyčejných stránek, které neobsahují žádná citlivá a důvěrná data, ale spíše by ho mohli využívat vedoucí projektů k vedení dokumentace, která je určena pouze určitému okruhu hostů.

Další výhodou je jednoduché přidělování statusu moderátora. To umožňuje administrátorovi rozdělit správu o systém mezi více uživatelů, aniž by tito uživatelé museli být administrátory.

Jako možné rozšíření připadá v úvahu detekce více druhů operačních systémů a prohlížečů. Můj program rozeznává 48 prohlížečů a 34 operačních systémů.

Literatura

[1] Wikipedia:Autentizace. http://cs.wikipedia.org/wiki/Autentizace.

[2] Wikipedia:Autorizace. http://cs.wikipedia.org/wiki/Autorizace.

[3] Wikipedia:Size of Wikipedia. http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia.

[4] Wikipedie:Historie Wiki. http://cs.wikipedia.org/wiki/Wiki, 2011-05-04 [cit. 2011-05-13].

[5] Aronsson, L.: General Purpose Wiki Website. http://aronsson.se/wikipaper.html, 2002-02-07.

[6] Brechlerová, D.: Bezpečnostní model založený na rolích a jeho realizace v XML security. http://dl.dropbox.com/u/26256434/917.pdf.

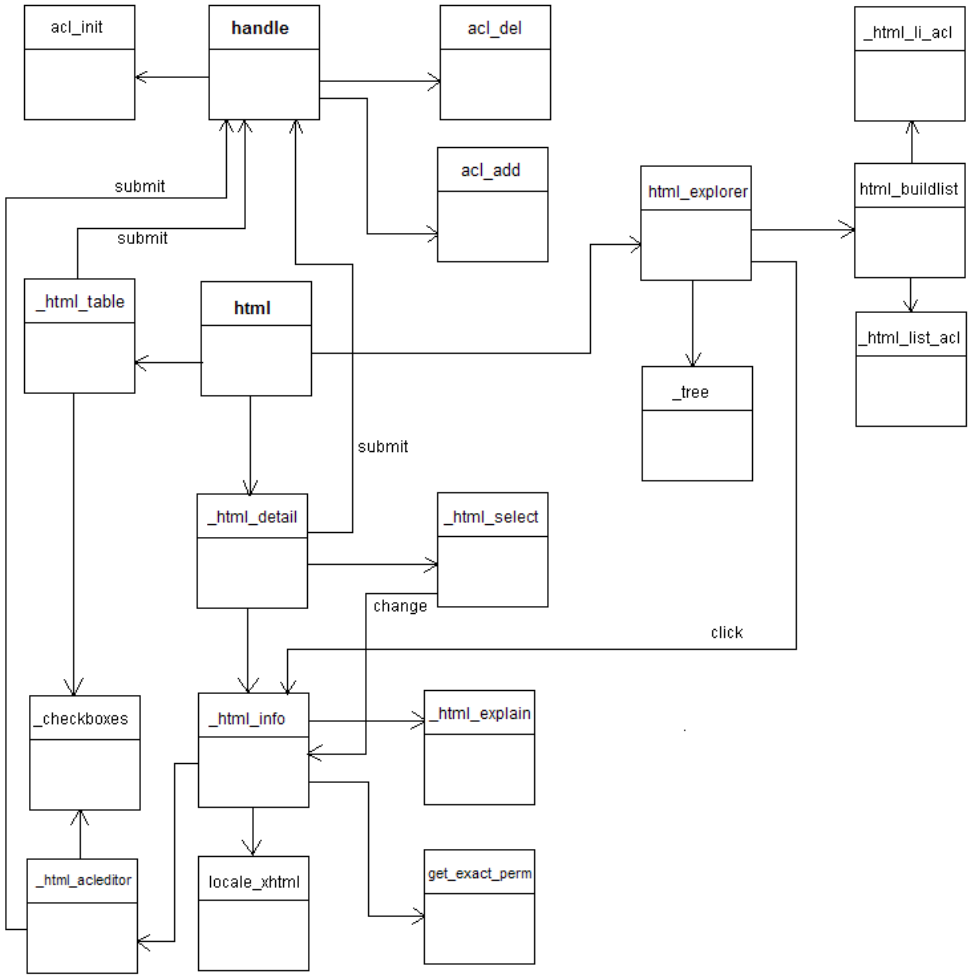
[7] Ferraiolo, D. F.; Kuhn, D. R.: *Role Based Access Control*. 1992, 15th National Computer Security Conference.

[8] Gohr, A.: Návod k používání DokuWiki. http://www.dokuwiki.org/dokuwiki, 2010-09-26.

[9] Sandhu, R. S.; Coyne, E. J.; Feinstein, H. L.: *Role-Based Access Control Models*. IEEE Computer 29(2), 1996, 38-47 s., iEEE Press.

Dodatek A

Callgraf funkcí jednoduchého přidělování práv



Obrázek A.1: Callgraf funkcí

Dodatek B

Obsah CD

Autlogin plugin

Vlastní zásuvný modul pro automatické přihlašování.

Textová část bakalářské práce

Latex soubory, které byly použity pro generování výsledného .pdf souboru.
Výsledný .pdf soubor s textem bakalářské práce.